

■ Flask File Upload MVC with MySQL, Logging, and Login Endpoint

Continuation of Midterm Activity

■ Overview

This activity continues your Flask File Upload MVC project from the midterm examination. You will enhance your application by adding a basic login endpoint that accepts a username and password and responds with a success or failure message. This version will not yet protect your file-upload routes; instead, it focuses on setting up and testing the login functionality as a foundation for future JWT integration.

■ Step 1: Review the Midterm Project

Ensure your previous project is complete and functional before beginning this task. Your project should include folders for controllers, models, services, uploads, and logs, as well as a working MySQL connection and proper logging setup.

■ Step 2: Create the Login Endpoint

Inside the controllers folder, create a new controller for authentication. It should accept a username and password, validate them (using static values for now), and return a response indicating login success or failure.

■ Step 3: Integrate the Login Controller into the Application

Ensure the login controller is registered in your main Flask app, similar to your upload controller, so that the login route becomes active and testable.

■ Step 4: Update Logging for Login Activity

Enhance your logging setup to include login attempts. Record each attempt—successful or failed—with details like username and result.

■ Step 5: Test the Login Endpoint Using Postman

Use Postman to send login requests and verify proper success and failure responses.

■ Step 6: Ensure Compatibility with Existing Endpoints

After adding the login feature, verify that all file upload CRUD operations still function correctly.

■ Step 7: Maintain Logging Consistency

Continue using your log directory and ensure all user activities—including login, upload, update, and deletion—are consistently recorded.

■ Step 8: Preparation for Future JWT Integration

Although authentication is not yet required, this login route will serve as the foundation for future JWT-based authentication. Later, you will use it to validate tokens and secure your endpoints.

■ Step 9: Submission Checklist

Before submission, verify that: the MVC structure is maintained; login endpoint works; logs are complete; all file upload functions remain intact; and the project runs smoothly without errors.

■ Summary

This task strengthens your MVC structure and prepares your Flask project for secure authentication. You will gain experience integrating endpoints, managing logs, and laying the groundwork for JWT security.